

Resum

El projecte es basa en el conjunt de publicacions del Departament de Matemàtica Aplicada I de la UPC sobre els subespais invariants. Els estudis del departament són entorn les deformacions que experimenten els objectes dels sistemes de control lineal, analitzant com varien les propietats i com resulta el sistema quan aquest és sotmès a petites pertorbacions. En particular, es treballarà amb els subespais invariants i, més específicament, els subespais marcats. Actualment s'està estudiant la classificació d'aquests subespais invariants, en la qual contribuïrem amb els resultats que s'obtinguin d'aquest projecte.

Treballarem amb les representacions dels subespais invariant mitjançant matrius BLD (*Block Lower Diagonal*), de les quals obtindrem la màxima simplificació. El projecte consisteix doncs, en l'obtenció d'un programa que permeti arribar de manera fàcil, ràpida i segura a la matriu BLD en la seva forma reduïda.

Així, en la primera fase del projecte es realitza un estudi de la teoria per obtenir la forma marcada maximal reduïda d'una matriu BLD i les possibles opcions que hi poden haver a l'hora de programar l'algoritme.

Posteriorment es mostra com s'utilitza el programa, dissenyat amb Office Excel, d'una manera senzilla, amb una interfície dissenyada per ser fàcil d'entendre i manipular. A continuació es detalla el funcionament de l'algorisme programat per a l'obtenció de les formes reduïdes, especificant l'estructura interna d'aquest per tal d'entendre el codi si es desitja modificar-lo.

Finalment s'analitzen els resultats que s'obtenen en aplicar l'algoritme per diferents formes de la matriu BLD, obtenint la deformació miniversal correcte i les formes reduïdes maximal marcades. S'observa com, per matrius BLD de mateixes dimensions però amb valors diferents, s'obté una mateixa forma reduïda maximal marcada.



Sumari

RESUM	1
SUMARI	3
GLOSSARI	5
PREFACI	6
Origen del projecte	6
Motivació	6
Requeriments previs	7
1. INTRODUCCIÓ	8
Objectius del projecte	8
Abast del projecte	8
2. SUBESPAIS INVARIANTS	10
3. SUBESPAIS MARCATS	12
4. REPRESENTACIÓ BLD D'UN SUBESPAI INVARIANT	14
5. TRANSFORMACIONS ELEMENTALS DE LA MATRIU BLD	18
6. FORMES MM	20
7. ANÀLISIS DE REQUERIMENTS	22
Demanda i estudi de mercat	22
Plantejament d'opcions	22
Llenguatge seleccionat i espai de treball	23
8. PROGRAMA, DISSENY I ESTRUCTURA	24
Plantejament del programa	24
Entorn de desenvolupament i llenguatge	24
Disseny del programa "Matriu BLD"	24
Resum d'entrades i sortides	29
Estructura del programa	30
9. RESULTATS	39
CONCLUSIONS	45
AGRAÏMENTS	47

BIBLIOGRAFIA _____ **48**

Referències bibliogràfiques 48

Bibliografia complementària 48



Glossari

F – cos que pot ser tant \mathbb{R} com \mathbb{C}

f – endomorfisme

E – espai vectorial sobre F de dimensió N

V – subespai vectorial de E

A – matriu o submatriu amb coeficients de F

M_n – matrius quadrades de n files i n columnes amb coeficients de F

p – partició $(p_1, p_2, p_3, \dots, p_n)$ de Jordan de A

q – partició $(q_1, q_2, q_3, \dots, q_m)$ de Jordan de la restricció de A a V

Y – matriu BLD dividida segons p en files i segons q en columnes

Y_{ij} – bloc de Y del bloc fila i i del bloc columna j

Y_{i*} – bloc fila de la matriu BLD

Y_{*j} – bloc columna de la matriu BLD

D_{ij}^h – diagonal h del bloc fila i i el bloc columna j

T_{ij}^k – triangles formats per les diagonals $D_{ij}^1, \dots, D_{ij}^k$

Prefaci

Per situar el marc del projecte i els seus objectius és necessari veure quin és el seu origen, la seva motivació i els requeriments previs. El projecte s'emmarca dins el conjunt d'investigacions del Departament de Matemàtica Aplicada I sobre els subespais invariants, es basa doncs, en els treballs previs realitzats pel departament.

Origen del projecte

El projecte ve originat pels estudis publicats pel Departament de Matemàtica Aplicada I sobre els subespais invariants, els estudis principals referència del projecte són la tesi de Marta Peña de deformacions de subespais [1], la tesi d'Albert Compta [2], els estudis dels subespais marcats d'Albert Compta, Josep Ferrer, Marta Peña i Ferran Puerta [3], així com altres publicacions de Josep Ferrer i Ferran Puerta com [4], de les deformacions versals dels subespais invariants.

L'objectiu principal del conjunt dels estudis del departament és estudiar les deformacions de diferents elements de la teoria de sistemes de control lineal per tal d'analitzar com varien les propietats qualitatives del sistema quan és sotmès a petites pertorbacions. En aquesta línia d'estudi apareix també la problemàtica de classificació dels elements dels sistemes que hi intervenen i en particular dels subespais invariants.

En l'estudi de les pertorbacions dels subespais invariants, però, no es coneix una forma canònica adequada, excepte per als subespais marcats que admeten bases de Jordan del subespai extensibles a l'espai total.

En la representació dels subespais invariants fem ús de la matriu *Block Lower Diagonal* (BLD), una matriu dividida per blocs formats per diagonals en el seu triangle inferior i zeros a la resta. Operant amb la matriu BLD, simplificant-la al màxim, podem estudiar la classificació del subespai invariant.

Motivació

La simplificació de les matrius BLD permet arribar a la forma Maximal Marcada (MM) del subespai invariant, la qual cosa facilita la seva classificació.

La motivació del projecte rau per tant en oferir un mitjà segur i senzill de simplificació de matrius BLD que permeti arribar a la forma reduïda MM per qualsevol subespai invariant. La

intenció és crear un programa que, donada una determinada configuració, executi les transformacions permeses en les matrius BLD fins arribar a la simplificació màxima. El programa ha de tenir una interfície fàcil d'utilitzar i d'interpretar per l'usuari, garantint que l'usuari no tingui la necessitat de conèixer la codificació interna d'aquest i pugui utilitzar-lo sense riscos d'errors.

Requeriments previs

Caldrà que el programa ofereixi una comprensió clara de les accions que es poden realitzar i les dades que s'han d'introduir. Podent operar amb matrius BLD de petita i gran dimensió, fent les transformacions necessàries per obtenir la seva simplificació màxima.

En l'estudi de les transformacions permeses de les matrius BLD apareixen restriccions determinades, les quals dificulten que les operacions siguin immediates i requereixen d'una programació exclusiva per aquest tipus de matrius. És per això que el programa ha de fer-se exclusivament per aquest cas i servirà només per a aquest estudi.

El programa ha de donar com a resultats la matriu BLD en la forma reduïda final així com el procés que s'ha seguit per arribar fins a aquesta.

1. Introducció

El projecte consisteix en l'elaboració d'un programa que ens permeti veure com es comporten unes determinades matrius en aplicar-hi transformacions concretes. Veurem com són aquestes matrius i quines són les transformacions que podem aplicar-les-hi. L'objectiu és obtenir resultats que confirmin la teoria dels marcats maximals, elaborada pel departament de Matemàtica Aplicada de la ETSEIB.

Objectius del projecte

L'objectiu del projecte és oferir una eina informàtica que permeti obtenir la forma reduïda MM d'una matriu BLD i el llistat de transformacions que s'han realitzat per obtenir-la. Amb l'objectiu final de poder seguir en l'estudi de la classificació dels subespais invariants del Departament de Matemàtica Aplicada I.

A partir del marcatge dels blocs de Jordan d'una matriu i la seva restricció es considera una matriu BLD i es desitja veure com es comporta davant les successives transformacions, segons les dimensions que la matriu pugui tenir, la configuració del marcatge de les matrius invariants i les diagonals principals marcades que es determinin. En particular s'obté la forma reduïda maximal marcada, amb diagonals principals marcades per la MM tipus.

Per fer-ho caldrà estudiar les transformacions permeses en les matrius BLD i configurar la programació necessària perquè puguin executar-se. En aquestes transformacions estudiarem com es comporten les matrius obtenint en particular la deformació miniversal de la tesi de Marta Peña [1], més específicament la MM forma reduïda.

Els objectius principals són, doncs, obtenir la deformació miniversal d'un subespai marcat i la forma reduïda maximal marcada de qualsevol subespai invariant representat en forma de matriu BLD.

Abast del projecte

Per poder realitzar la programació cal fer un estudi previ de la teoria de les deformacions dels subespais marcats a partir de la qual es parteix i les transformacions que permeten arribar a la deformació miniversal. És per això que primerament s'estudiaran els subespais marcats, i a continuació s'iniciarà en un estudi detallat de les matrius BLD i les transformacions permeses.

Un cop obtingudes detalladament les diferents transformacions que es poden aplicar caldrà veure quines són les necessitats i com es podrà programar l'algoritme que permeti solucionar-ho. Caldrà fer un petit estudi de les diferents possibilitats de programació que existeixen, avaluant-les i decidir amb quina s'operarà. En el cas que la programació requereixi de coneixements no adquirits caldrà iniciar un procés d'iniciació en el llenguatge de programació pertinent.

Seguidament es programarà tot l'algoritme de les transformacions de la matriu BLD, la obtenció de la deformació miniversal i la obtenció de la forma reduïda maximal marcada. Es descriurà tota l'estructura interna del programa per a la comprensió d'aquest, de manera que es garanteixi una comprensió fàcil de l'ús i del funcionament del programa. I finalment s'hi afegirà una descripció més detallada que permeti conèixer la metodologia del codi per si es desitja modificar-lo.

Un cop realitzat el programa es duran a terme diverses proves per veure quins són els resultats que s'obtenen de diverses configuracions d'una mateixa matriu BLD inicial.

2. Subespais invariants

En tota la memòria suposarem E un espai vectorial sobre \mathbb{C} , de dimensió N i $f: E \rightarrow E$ una aplicació lineal.

Definició 2.1. Un subespai $V \in E$ se'n diu *f-invariant* si $f(v) \in V$ per a tot vector $v \in V$. Aleshores escriurem $\hat{f}: V \rightarrow V$ com la restricció de la f a V . Si A és la matriu de f en una base qualsevol, direm també que V és *A-invariant*.

Com veurem de seguida la representació d'un subespai invariant dependrà de les bases triades a V i a E . Per tal d'agrupar els que poden tenir una mateixa representació (amb una adequada tria de bases a V i a E), s'introdueix la següent relació d'equivalència.

Definició 2.2. Dos subespais A -invariants V i V' se'n diuen *equivalents* si existeix una matriu de canvi de base S tal que:

$$S^{-1} A S = A', S^{-1} V = V'$$

La caracterització d'aquestes classes d'equivalència és encara un problema obert. Per tal d'avançar en aquesta direcció a [5] s'introdueix una representació especialment simple de cada subespai invariant, anomenada *maximal marcada* (MM). L'objectiu d'aquest treball és computar de forma eficient aquesta representació.

Es demostra fàcilment que l'estudi de subespais invariants es pot reduir al cas de f nilpotent, amb una forma de Jordan predeterminada, que caracteritzarem per la grandària dels seus blocs:

$$p_1 \geq \dots p_i \geq \dots \geq p_n > 0, p_1 + \dots + p_n = N$$

Igualment podrem predeterminar la dimensió d de V i la grandària dels blocs de Jordan de la restricció \hat{f} :

$$q_1 \geq \dots q_j \geq \dots \geq q_m > 0, q_1 + \dots + q_m = d$$

Indicarem per $Inv(p, q)$ aquests subespais invariants, on:

$$p = (p_1, \dots, p_n), q = (q_1, \dots, q_m)$$

És freqüent representar p i q mitjançant diagrames de Young, com s'il·lustra en l'exemple 1.1. Les cadenes de Jordan es representen com descendents.

Exemple 1.1: en tota la memòria emprarem sovint l'exemple 1.2.9 de la tesi de Marta Peña [1]:

$$p = (6, 4, 3, 2), q = (4, 2, 1)$$

De les quals obtenim els diagrames de Young respectius:

$$p_1 = 6$$

u_1			
Au_1	$p_2 = 4$		
\downarrow	u_2	$p_3 = 3$	
	\downarrow	u_3	$p_1 = 2$
		\downarrow	u_4
			\downarrow

$$q_1 = 4$$

v_1		
\downarrow	$q_2 = 2$	
	v_2	$q_3 = 1$
	\downarrow	v_3

3. Subespais marcats

Una classe de subespai invariant especialment important és la dels anomenats marcats. Són aquells que admeten una base de Jordan relativa a la restricció que sigui extensible a una base de Jordan de l'espai total corresponent. Els subespais marcats juguen un paper clau en el nostre treball.

Podem veure a estudis previs tals com la tesi d'Albert Compta [2] les caracteritzacions geomètriques dels subespais marcats:

Definició 3.1. Un subespai V es diu f -marcat si existeix una base de Jordan de V respecte a la restricció \hat{f} que pot estendre's a una base de Jordan de E relativa a f .

És trivial que tot f -marcat és f -invariant i que tot subespai equivalent a un marcat també és marcat.

Exemple 3.1: Suposem $p = (3,1)$ i una base de Jordan:

$$u_1, Au_1, A^2u_1; u_2$$

Per a $q = (2)$ és:

$$V_1 = [Au_1, A^2u_1] \in \text{Inv}(p, q)$$

$$V_2 = [Au_1 + u_2, A^2u_1] \in \text{Inv}(p, q)$$

V_1 és marcat, mentre que V_2 no ho és ja que no té una base de Jordan adaptada.

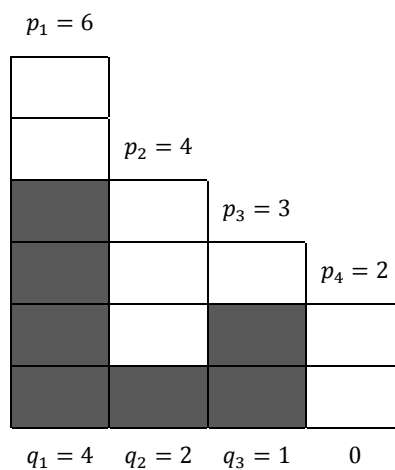
El diagrama de Young d'un subespai de $\text{Inv}(p, q)$ marcat és un subdiagrama del de p , amb reordenacions adequades. Per una configuració de q hi ha diferents combinacions possibles, a partir de les quals es poden classificar els subespais marcats.

Exemple 3.2: per a $p = (6, 4, 3, 2)$ i $q = (4, 2, 1)$, el subespai marcat:

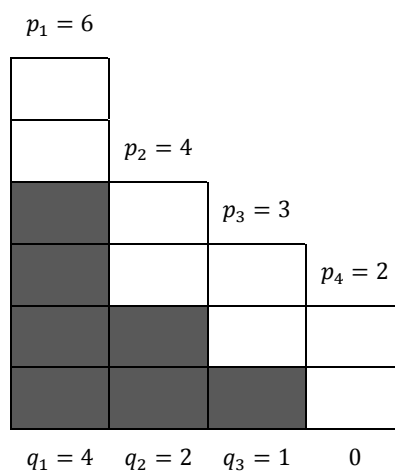
$$V = [A^2u_1, A^3u_1, A^4u_1, A^5u_1; A^3u_2; Au_3, A^2u_3]$$



podem representar-lo com:



O alternativament



4. Representació BLD d'un subespai invariant

Per representar un subespai V f -invariant es prenen per columnes les coordenades d'una base de Jordan de V en una base de Jordan de f , resultant una matriu de tipus BLD que precisem a continuació.

Definició 4.1. Direm que una matriu Y és BLD (Block Lower Diagonal) si és una matriu per blocs on cada un dels blocs és una matriu triangular inferior constant al llarg de les diagonals. En particular direm que és $BLD(p, q)$ si la partició per files és $p = (p_1, p_2, \dots, p_n)$ i per columnes $q = (q_1, q_2, \dots, q_m)$. Doncs, el bloc Y_{ij} (on i indica fila i j indica columna) té p_i i q_j columnes. Indicarem per $BLD^*(p, q)$ les que tinguin rang màxim ($= d$).

L'ordenació de p és de dalt a baix i la de q d'esquerra a dreta, tal com il·lustra l'exemple 4.1.

Exemple 4.1. Per a $p = (6, 4, 3, 2)$ i $q = (4, 2, 1)$, la matriu $BLD(p, q)$ és de la forma

		$q_1 = 4$	$q_2 = 2$	$q_3 = 1$	
$p_1 = 6$		*			
		*	*		
		*	*	*	
		*	*	*	*
			*	*	*
$p_2 = 4$		*			
		*	*		
		*	*	*	
		*	*	*	*
			*	*	*
$p_3 = 3$		*			
		*	*		
		*	*	*	*
			*	*	*
$p_4 = 2$		*			
		*	*		
			*	*	*



Per poder seguir detallant la matriu BLD precisem algunes notacions addicionals.

Definició 4.2. Anomenarem:

1) Bloc-fila: $Y_{i*} =$

y_{i1}	\dots	y_{im}
----------	---------	----------

essent i l'indicador del bloc fila.

2) Bloc-columna: $Y_{*j} =$

y_{1j}
\dots
y_{nj}

essent j indicador del bloc columna.

3) Per a cada bloc Y_{ij} les diagonals es numeren, començant per baix-esquerra: $D_{ij}^1, \dots, D_{ij}^k, \dots; 1 \leq k < p_i + q_j$. Notarem els seus coeficients com y_{ij}^k .

Clarament $y_{ij}^k = 0$ si $k > \min(p_i, q_j)$. Per a les altres direm que tenen *altura* (o llargària) k i *profunditat* $p_i - k$.

4) Les diagonals-fila D_{i*}^h i les diagonals-columna D_{*j}^h són, respectivament:

D_{i1}^h	\dots	D_{im}^h

D_{1j}^h
\dots
D_{nj}^h

- 5) Distingim diferents tipus de diagonals que podem trobar: *diagonal maximal*, la no nul·la de màxima altura; *diagonal principal*, la d'altura q_j , és a dir, $D_{ij}^{q_j}$, nul·la si $q_j > p_i$.
- 6) Anomenem la *submatriu principal* de Y la formada per $y_{ij}^{q_j}, 1 \leq i \leq n, 1 \leq j \leq m$.
- 7) Escriurem T_{ij}^k els triangles formats per les diagonals $D_{ij}^1, \dots, D_{ij}^k$.

El nostre interès en les matrius $BLD(p, q)$ rau en la proposició següent, de fàcil comprovació:

Proposició 4.1:

- 1) Sigui $V \in \text{Inv}(p, q)$. Si disposem per columnes les coordenades d'una base de Jordan de V qualsevol, respecte a una base de Jordan de E qualsevol, resulta una matriu $Y \in BLD^*(p, q)$. Recíprocament, si $Y \in BLD^*(p, q)$, el subespai generat per les seves columnes és de $\text{Inv}(p, q)$.
- 2) Una matriu $Y \in (p, q)$. té rang màxim ($= d$) si i només si la seva submatriu principal té rang màxim ($= m$).

Doncs, els subespais de $Inv(p, q)$ poden representar-se per matrius de $BLD^*(p, q)$.

Exemple 4.2. El subespai V de l'exemple anterior pot representar-se per

$Y =$

1						
	1					
		1				
			1			
						1
				1		
					1	

En general, un subespai marcat pot representar-se per una matriu $Y \in BLD^*$ amb tots els coeficients 0 excepte una diagonal principal a cada bloc-columna (que podem suposar de 1), situades en diferents blocs-files

Aquesta representació Y no és única ja que és ben conegut que:

Proposició 4.2.. Sigui $Y \in BLD^*(p, q)$ una representació de $V \in Inv(p, q)$

1) Multiplicar Y per la dreta per $Q \in BLD^*(q, q)$ equival a fer un canvi de base de Jordan de V . Doncs, les altres representacions BLD de V són:

$$YQ, Q \in BLD^*(q, q)$$

2) Multiplicar Y per l'esquerra per $P \in BLD^*(p, p)$ equival a fer un canvi de base de Jordan



de E . Doncs, les representacions de tots els subespais equivalents a V són:

$$PYQ, P \in BLD^*(p, p), Q \in BLD^*(q, q)$$

Per tant, per facilitar l'estudi dels subespais $Inv(p, q)$, cal buscar P i Q com abans per tal que PYQ sigui el més senzilla possible. En el nostre cas, a [5] es troben les que anomenem formes reduïdes MM (maximal marcada). Aquí computarem una d'aquestes MM formes reduïdes.

5. Transformacions elementals de la matriu BLD

L'algoritme per obtenir una Maximal Marcada (MM) en forma reduïda de $Y \in BLD^*(p, q)$ es basa en el fet que multiplicar per P i Q com abans equival a “pivotar” per blocs-fila o per blocs-columna. Per tal de simplificar, doncs, la matriu BLD es fa ús dels mètodes de pivot per files i pivot per columnes, amb les restriccions que s'especifiquen tot seguit.

Definició 5.1 Una transformació per files o columnes és *permesa* si i només si conserva l'estructura de la matriu $BLD(p, q)$.

Equivalentment a multiplicar per la dreta per $BLD^*(q, q)$ o per l'esquerra per $BLD^*(p, p)$, les transformacions permeses són:

Proposició 5.1. [5] Sigui $Y \in BLD(p, q)$:

- 1) Els productes YQ , $Q \in BLD^*(q, q)$ poden obtenir-se mitjançant les següents transformacions elementals:
 - a. Multiplicar tot un bloc-columna per un escalar diferent de 0.
 - b. Sumar les k darreres columnes d'un bloc-columna a les k primeres d'un altre o d'ell mateix.
- 2) Els productes PY , $P \in BLD^*(p, p)$, poden obtenir-se mitjançant les següents transformacions elementals:
 - a. Multiplicar tot un bloc-fila per un escalar diferent de 0.
 - b. Sumar les k files superiors d'un bloc-fila a les k inferiors d'un altre o d'ell mateix.

Combinant adequadament les transformacions anteriors, podem “pivotar per diagonals” com segueix:

A partir d'una diagonal D_{ij}^h maximal es pretén eliminar una altra, per fer-ho es pivota prenent totes les files o columnes de l'altura h de la diagonal sobre el bloc desitjat i es pivota successivament prenent $n - 1, n - 2, \dots, 1$ files o columnes.

Ens interessa també poder seleccionar el tros de la diagonal maximal (cua de diagonal maximal) adient per pivotar i eliminar-ne alguna altra d'igual o menor altura i netejar els valors afectats per altres transformacions. De tal manera que es netegin automàticament els valors afectats.



Corol·lari 5.1. Sigui $D_{ij}^{q_j}$ una diagonal principal no nul·la.

- 1) Podem pivotar-la per columnes sobre les del seu mateix bloc Y_{ij} i sobre les altres $D_{ij'}^k$ amb $k \leq q_i$ (és a dir, de menor altura).
- 2) Podem pivotar-la per files sobre les del seu mateix bloc Y_{ij} i sobre les altres $D_{i'j}^k$ amb $k \leq q_j$ i $p_i - k \geq p_i - q_j$ (és a dir, de menor altura i major profunditat).

Observacions:

- 1) Cal alertar que, en fer (1), altres diagonals $D_{ij'}^k$ també pivoten. Concretament si $D_{ij}^{q_j+\Delta} \neq 0$, també resulten afectades $D_{ij'}^{\Delta+1}, \dots, D_{ij'}^{\Delta+k}$. I anàlogament per a (2).
- 2) L'algoritme ha de donar no només el resultat d'aplicar (1) i (2) reiteradament sinó també les matrius P i Q que igualment el donin.

Exemple 5.1. Emprant (1) i (2) podem obtenir la deformació miniversal d'una matriu marcada construïda a [1]. Addicionalment obtenim les matrius P i Q que la donin.

Partint de $Y_0 \in BLD^*(p, q)$ marcada i considerant pertorbacions $Y_0 \in BLD^*(p, q)$ de tots els seus coeficients excepte els inicials de Y_0 , es tracta de fer nul·les el màxim possible d'aquestes pertorbacions pivotant adequadament, fins arribar a reduir-se a l'anomenada deformació miniversal:

- i) Seleccionant els blocs principals inicials del més baix al més alt, pivotarem amb la seva diagonal principal per files cap amunt
- ii) A continuació, seleccionant els blocs principals en l'ordre invers, pivotarem per columnes a dreta i esquerra.
- iii) Finalment, altre cop en l'ordre de (i) pivotarem per files cap avall.

La selecció dels blocs principals i l'ordre a de les transformacions és fonamental per a preservar els 0 que anirem obtenint (vegeu (1) de l'observació anterior).

6. Formes MM

A l'hora de trobar una representació BLD senzilla d'un subespai $V \in \text{Inv}(p, q)$, el punt de partida és que tot subespai invariant és pertorbació d'un marcat, de manera que representacions seves apareixeran en les deformacions miniversals de marcats que acabem de referir. Tanmateix, un mateix subespai pot estar representat en diverses deformacions miniversals. A [5] es dóna un mètode per a seleccionar-ne una d'elles de forma única, anomenant-se *tipus maximal marcat* (MM) el subespai marcat corresponent. Addicionalment, es minimitza el nombre de paràmetres no nuls en la deformació miniversal, cosa que justifica l'apel·latiu de "maximal". En direm *formes reduïdes maximals marcades* (MM) les representacions d'un $V \in \text{Inv}(p, q)$ que apareixen en aquesta subfamília de la deformació miniversal del seu tipus MM.

L'algorisme següent permet obtenir, a partir d'una representació $Y \in \text{BLD}^*(p, q)$ qualsevol d'un subespai $V \in \text{Inv}(p, q)$, el seu tipus MM i una forma reduïda Y_{MM} . Aquí implementarem aquest algorisme.

Teorema 6.1. [5]. Sigui $V \in \text{Inv}(p, q)$ i $Y \in \text{BLD}^*(p, q)$ una representació seva qualsevol.

- 1) Per a $j = 1$, seleccionarem el bloc $Y_{i_1 1}$ més avall amb la diagonal principal no nul·la. Pivotarem per files cap amunt amb $D_{i_1 1}^{q_1}$, de manera que resulti la única diagonal principal no nul·la en el bloc columna Y_{*1} , que podem suposar 1 dividint tot el bloc-columna. Pivotarem per columnes anul·lant les diagonals principals de la seva dreta dins el seu bloc fila.

Procedirem anàlogament per $j = 2, 3, \dots, m$, de manera que finalment a cada bloc-columna $Y_{*1}, Y_{*2}, \dots, Y_{*m}$ hi ha exactament una diagonal principal no nul·la (que podem suposar 1), en els blocs $Y_{i_1 1}, \dots, Y_{i_m m}$, situats en blocs fila diferents.

Els índex obtinguts i_1, \dots, i_m no depenen de la representació inicial Y de V .

- 2) A partir de les diagonals principals seleccionades procedirem com a l'exemple 5.1, obtenint ara la forma reduïda maximal marcada:
 - i) Seleccionant els blocs principals del més baix al més alt, pivotarem amb la seva diagonal principal per files cap amunt
 - ii) A continuació, seleccionant els blocs principals en l'ordre invers, pivotarem per columnes a dreta i esquerra.
 - iii) Finalment, altre cop en l'ordre de (i) pivotarem per files cap avall



Definició 5.1.. En les condicions anteriors, el subespai marcat corresponent als índex i_1, \dots, i_m se'n diu el *tipus maximal marcat* (MM) de V , i la representació obtinguda a (2), la seva *forma reduïda maximal marcada* (MM).

Observacions:

- 1) La forma reduïda MM obtinguda a (2) no és única, sinó que depèn de la representació inicial Y .
- 2) Totes les formes reduïdes MM apareixen en una subfamília de la deformació miniversal del tipus MM: la que resulta de fer nuls els paràmetres de les diagonals principals diferents de les corresponent al tipus MM.

7. Anàlisi de requeriments

Demanda i estudi de mercat

Degut a les característiques de l'estudi no existeix cap programa que resolgui les operacions desitjades. Existeixen opcions com l'Excel o el Matlab que permetrien operar amb la matriu BLD. El Matlab permet operar amb matrius, multiplicant per la dreta o per l'esquerra. L'Excel opera per taules, podent operar per columnes, files o caselles. Però l'estudi requereix una sèrie d'instruccions determinades que han d'executar-se reiteradament, per tant és necessària l'elaboració d'un codi programable que executi les diferents operacions, tenint en compte totes les restriccions.

Tant el Matlab com l'Excel permeten programar funcions i executables, però vegem el conjunt d'ofertes que tenim per codificar l'algoritme.

El primer objectiu de l'estudi és poder operar amb la matriu BLD per obtenir la seva deformació miniversal i la seqüència dels passos necessaris que s'han fet per arribar-hi. Un cop estudiades les possibles operacions permeses i les respectives restriccions podem procedir a descriure la demanda en detall.

El programa haurà de tenir les següents característiques:

- Fàcil utilització, és a dir que es pugui fer servir en els sistemes operatius habituals, preferiblement Windows.
- Memòria interna, per suportar càlculs de grans dimensions
- Versatilitat, per programar funcions amb restriccions
- Estructura en matrius o taules
- Interfícies de sortida de dades que permetin la visualització de la matriu BLD en la seva forma reduïda.

Plantejament d'opcions

El principal llenguatge de programació dels estudis del Grau d'Enginyeria en Tecnologies Industrials és el *Python*, amb el qual es poden programar tota la sèrie de funcions i classes del conjunt de la codificació necessària. També es pot obtenir una interfície de sortida en forma de gràfic, però és més difícil si aquesta ha de ser alhora una matriu de valors dins de la gràfica. Tindria l'opció també de mostrar la matriu en forma de text. El problema és que caldria estudiar bé com es podria fer per habilitar el programa a l'ús en Windows.



Pel Departament de Matemàtica Aplicada és més comú el Matlab, una plataforma optimitzada per resoldre problemes d'enginyeria i ciències. El llenguatge de Matlab està basat en matrius, amb gràfics integrats i permet programar funcions. Les interfícies de sortida però són limitades, i al haver d'operar amb tantes restriccions no podem operar amb les matrius directament sinó que hem de crear les nostres pròpies funcions.

Les operacions permeses estan basades en operacions de certes columnes i certes files, que s'han de recórrer i actualitzar-se. L'Excel és d'ús comú i permet operar amb una interfície en forma de taula de valors, en files i columnes. L'aplicació Excel forma part del conjunt Microsoft Office i té com a finalitat d'ús els fulls de càlcul per ús econòmic, financer i comptable. Inclou també l'eina dels Macros, que permeten programar, gravar i executar accions repetidament quan es desitgi i quan sigui necessari. Aquests macros permeten crear funcions, interfícies d'entrades de dades i actuen sobre el full de càlcul que es desitgi. El llenguatge de programació dels Macros és el Visual Basic.

Degut a la facilitat d'ús, la interfície de sortida en forma de taula i la facilitat de programar les diferents funcions l'opció escollida és l'Excel. Aquesta opció requerirà una fase prèvia d'aprenentatge del llenguatge Visual Basic.

Llenguatge seleccionat i espai de treball

El llenguatge de Visual Basic és un llenguatge de Microsoft per a dissenyar les funcions dels Macros, dins de l'entorn de desenvolupament es pot executar el programa i també es pot guardar en format .exe, essent possible la seva execució sense tenir l'aplicació oberta. Existeixen diferents versions disponibles del llenguatge, des de la 2.0 fins a la 20.0, amb algunes variacions en cada un.

A l'espai de treball es diferencien tres tipus d'objectes, els Llibres, els Fulls i els Gràfics. En els Fulls hi ha les caselles, les files, les columnes i els rangs. Tots aquests elements són objectes, és a dir que tenen propietats que els descriuen i mètodes o accions que poden realitzar.

8. Programa, disseny i estructura

Plantejament del programa

L'objectiu del programa és simplificar al màxim la matriu BLD per obtenir-ne la seva forma reduïda. Per fer-ho es pot fer operació a operació, escollint quina utilitzar a cada pas, o es pot fer sistemàticament seguint una seqüència determinada. L'objectiu final és poder obtenir el llistat de les operacions realitzades, la deformació miniversal resultant, la MM tipus i la MM forma, tant si s'ha fet pas a pas com si s'ha fet seqüencialment.

L'obtenció de la forma reduïda però no verifica que la solució sigui la única, per verificar-ho seria necessari realitzar un altre programa que definís l'arbre de marcats.

Entorn de desenvolupament i llenguatge

Per codificar el programa s'ha fet ús del llenguatge Visual Basic en els Macros de l'Excel, un programa de Microsoft Office que té l'opció de treballar amb codi d'execució. Els macros ens permeten automatitzar tasques d'ús quotidià o repetit, de manera que l'ús del programa sigui més eficient. Un macro són una sèrie de comandes o instruccions que queden emmagatzemades dins del document d'Excel i que es poden executar sempre que sigui necessari o es desitgi.

En el nostre cas el conjunt de codi programat es troba a l'arxiu *Programa Matriu BLD*, les línies de codi es poden consultar obrint el Desenvolupador de l'Excel. Per defecte l'opció *Desenvolupador* no apareix així que cal configurar-la, o clicar Alt+F11.

Accedint al Desenvolupador es poden consultar els diferents elements que constitueixen el programa, però ens interessa que l'usuari només hagi de fer ús de la interfície habitual de l'Excel, és a dir el llibre de càlcul amb els seus respectius fulls de càlcul.

Disseny del programa "Matriu BLD"

El programa està dissenyat per un ús senzill i el màxim entenedor possible, però per fer-lo servir cal conèixer el comportament de la matriu BLD i com obtenir la matriu en la seva forma reduïda. Es disposa d'una taula d'instruccions amb un seguit de botons ordenats estratègicament per poder seguir una seqüència ordenada. El primer pas serà crear la matriu amb les dimensions desitjades i posteriorment se li poden realitzar transformacions pivot o



també aplicar algoritmes seqüencials que porten més fàcilment a la forma reduïda.

Cada un dels botons està associat a un macro que executa les funcions corresponents.

Introducció de la matriu BLD(p,q)

En el llibre de comandes de l'Excel apareixen una sèrie de botons amb les diferents accions que es poden realitzar, la primera és la de creació de la matriu, la qual inicia el macro *agregar* que llença un *UserForm* per introduir valors.

Els *UserForm* són formularis d'introducció de dades dissenyats exclusivament per cada botó d'execució i que farem servir per introduir les diferents dades.

En fer clic a *Construir matriu BLD* s'inicia el programa, preparant l'espai de treball. Es crea un segon llibre d'Excel anomenat *Matriu_BLD* amb tres fulls anomenats *Matriu BLD*, *Valors p i q* i *Llista d'operacions*. En el primer full es mostrarà la matriu BLD sobre la que s'anirà operant, al segon full diversos valors que s'introduiran i al tercer s'emmagatzemarà la informació de les operacions realitzades.

Cal introduir els valors de p i q , de cada columna de l'edifici de Jordan, per ordre, d'esquerra a dreta. Quan no s'introdueix un valor de p no executa res, quan no s'introdueix cap valor de q l'entén com a 0. En cas que la dimensió de q sigui superior a la de p dona error de dimensió, ja que els valors de q no poden ser superiors als de p .

Un cop introduïts tots els valors, fent clic a *Fi*, apareix la matriu BLD amb les files i columnes definides, i només resta introduir els valors de les diagonals.

El programa emmagatzema els valors de p i q en les dues columnes del segon full del llibre, essent aquests accessibles segons l'índex $j = 1,2,3 \dots n$ i $i = 1,2,3 \dots m$ essent l'1 el primer valor introduït, n el darrer de les files i m el darrer de les columnes.

El següent botó *Introduir valors* és per introduir els valors de les diagonals, en fer clic s'inicia el macro *introduir* que pinta les caselles on s'han d'introduir els valors i automàticament els copia a les diagonals respectives. Es bloquegen també la resta de caselles, per a evitar errors



Imatge 1 Taula d'instruccions

de teclejat, però si aquestes es volen desbloquejar es pot seleccionar tot el full i clicar *Revisar: Desprotegeix el full*.

En el cas de treballar amb matrius molt grans és permès treballar visualitzant únicament la fila inferior de cada bloc, simplificant així la visualització de les matrius de grans dimensions. Les operacions no es veuen afectades, ja que totes elles es realitzen exclusivament a partir de la última fila dels blocs, la resta de valors són iguals als de les diagonals corresponents. Fent clic a *Forma reduïda* es visualitzen només les últimes files dels blocs fila, hi segueixen sent, però l'Excel no les mostra. Per tornar a visualitzar la matriu sencera només cal fer clic al botó *Forma completa*.

En cas de voler tornar al pas anterior clicant a *Pas anterior* la matriu BLD es restaura segons la situació abans de la última operació i esborra les operacions realitzades en el full 3. Només permet, però, un pas enrere.

Llistat d'operacions permeses

- Pivotar per columnes:

Fent clic al botó *Pivotar per columnes* s'inicia el macro *crida_pc* que llença un formulari per introduir valors. Cal seleccionar el bloc-columna, indicar el nº de columnes amb les que s'operarà (es seleccionaran les de més a la dreta del bloc), indicar per quin escalar es multipliquen (si no s'introdueix s'entén com a 1), i el bloc-columna al que van a sumar-se (es sumaran a les columnes de més a l'esquerra, tantes com se n'han seleccionat, en l'ordre original). En cas que l'operació no estigui permesa el programa retorna error, és a dir, en el cas que l'escalar sigui 0. Retorna error també en cas que el nº de columnes sigui major que la dimensió d'algun dels blocs indicats, o que alguna de les columnes indicades estigui fora de la dimensió de la BLD.

- Pivotar per files

Fent clic al botó *Pivotar per files* s'inicia el macro *crida_pf* que llença un formulari per introduir valors. Cal seleccionar el bloc-fila, indicar el nº de files amb les que s'operarà (es seleccionaran les files superiors del bloc), indicar per quin escalar es multipliquen (si no s'introdueix s'entén com a 1), i el bloc-fila al que van a sumar-se (es sumaran a les columnes més inferiors, tantes com se n'han seleccionat, en l'ordre original). En cas que l'operació no estigui permesa el programa retorna error, és a dir, en el cas que l'escalar sigui 0. Retorna error també en cas que el nombre de files sigui major que la dimensió d'algun dels blocs indicats, o que alguna de les files indicades estigui fora de la dimensió de la BLD.



- Multiplicació d'una fila o d'una columna

Els respectius botons llencen formularis d'introducció de dades, en els que s'introdueix la fila o columna i l'escalar (si no s'introdueix s'entén com a 1). Dóna errors si l'escalar és 0 o si la fila o columna indicada és major que la dimensió de la BLD.

- Pivotar per diagonals

Fent clic a *Pivotar per diagonal maximal* s'inicia el macro *pivot_dm* que llença un formulari per introduir valors. Cal seleccionar la diagonal D_{ijh} a partir de la qual s'actuarà, seleccionar el tipus d'operació que es vol, si per columnes (dreta o esquerra) o per files (amunt o avall), indicar per quin escalar es multiplica (si no s'introdueix s'entén com a 1), i sobre quina diagonals D_{ijh} actua. En seleccionar l'operació per columnes s'inhabilita l'opció d'indicar la fila de destí, i s'inhabilita la de columna si es selecciona l'operació per files, ja que aquests valors no es necessiten, es mantenen. El programa guarda els valors actuals de la última fila destí per tal d'operar posteriorment sobre ells i acabar obtenint els mateixos. Un cop emmagatzemats pivota per columnes o per files sobre tot el bloc amb el número de files o columnes igual a la n dimensió de la diagonal maximal. Posteriorment crida successivament la funció *Pivot per columnes* prenent $n - 1, n - 2, \dots 0$ files o columnes amb l'escalar adequat per obtenir els valors guardats inicialment, fent ús de la fórmula:

$$valor\ desitjat = valor\ actual + escalar * valor\ diagonal\ maximal \quad (1)$$

En cas que l'operació no estigui permesa el programa retorna error, és a dir, en el cas que l'escalar sigui 0 o en el cas que alguna de les files o columnes indicades estigui fora de la dimensió de la BLD.

Cal tenir en compte que quan es fan aquestes operacions, en tots tres casos, s'arrossegueu les operacions a tot el bloc fila o columna.

Tot i que el formulari no ho permet, la funció està habilitada per operar amb només una part de la diagonal, essent possible doncs l'operació amb només un tram de la diagonal principal. Quan no s'introdueix un valor de dimensió desitjat s'entén que s'opera amb tota la diagonal sencera. Aquest mètode s'utilitza en la neteja per transformacions successives.

- Transformacions pivot successives:
 - En una direcció determinada

L'opció *Neteja segons direcció* permet introduir la coordenada fila i la coordenada columna del bloc desitjat i la direcció en la que es vol operar. Només està permès seleccionar una de les opcions de direcció. En clicar *Neteja* el programa inicia el macro *dreta, esquerra, amunt o avall*, segons l'opció indicada. Cada una d'aquestes opcions crida la funció de *Pivotar per files*

o *Pivot per columnes* per totes les successives files o columnes en les que pot operar, tant en ella mateixa en el cas de *dreta* o *esquerra*, i en la resta per totes. Opera a partir de la diagonal principal seleccionada, agafant la dimensió d'aquesta necessària per operar en cada bloc i netejar cada un dels elements. Només opera quan el valor de la diagonal a la que operem es diferent de 0, ja que sinó, com veiem a la fórmula (1) no podem trobar el valor de l'escalar necessari.

- En totes direccions, en tota la matriu BLD

Per a poder netejar tota la matriu és necessari seleccionar les diagonals principals a partir de les quals volem “miniversalitzar”, així doncs el programa demana les coordenades dels blocs on es troben les diagonals principals amb un formulari d'entrada de dades. Mentre s'introdueixen les dades a través del formulari es mostren els valors que es van introduint al segon full del llibre *Matriu_BLD*. Per guardar els valors els escriu en dues columnes, una per les coordenades fila i una per les columnes, anomena el conjunt com a *Selecció de diagonals principals*. Les diagonals poden introduir-se en ordre aleatori sense afectar l'execució de l'algoritme. No pot introduir-se una diagonal principal d'una fila ni d'una columna que ja s'hagi seleccionat, segons l'error que es produeixi el programa indica què l'ha causat.

A partir d'aquestes diagonals introduïdes fent clic a *Fi* el programa crida la funció *miniversal* que executa l'algoritme de neteja que inicia les transformacions, primer pivotant per files amunt, de la més avall a la més amunt, després per columnes dreta i esquerra, de la fila més amunt a la més avall, i altre cop per files, avall, de la fila més avall a la més amunt.

En cada operació realitzada es mostra la matriu BLD actualitzada i s'anota l'operació al tercer full del llibre d'Excel, amb el nº de l'operació, el tipus, la dimensió, la nomenclatura de la dimensió i la fórmula de l'operació.

Eliminar

L'opció *Eliminar* tanca el llibre “Matriu_BLD”, donant l'opció a guardar-lo si es desitja. Si el botó de tancar falla després d'executar alguna part del programa és segur fer servir aquesta opció.



Resum d'entrades i sortides

A continuació s'indica un breu resum pas a pas per utilitzar el programa fàcilment:

1. S'introdueix la llista p i la llista q , formades pel nombre de pisos de cada columna de la representació segons el model de Young de la matriu de Jordan sobre la que volem operar (p) i els de la restricció sobre la que operem (q). Cada vegada que s'entra un valor de p i un valor de q (opcional) s'ha de clicar *Introduir més valors*.
2. S'introdueixen els valors de les diagonals a la fila inferior de cada bloc, a les caselles marcades.
3. S'indica el tipus d'operació que volem realitzar i per a cada cas s'introdueix:

Taula 1 Resum d'entrades

Operació	Introduir
<u>Pivot per columnes</u>	<ul style="list-style-type: none"> ▪ Bloc-columna j a partir del qual actuem ▪ Nombre de columnes desitjat ▪ L'escalar pel que es multipliquen ▪ Bloc-columna sobre el qual actuem
<u>Pivot per files</u>	<ul style="list-style-type: none"> ▪ Bloc-fila i a partir del qual actuem ▪ Nombre de files desitjat ▪ L'escalar pel que es multipliquen ▪ Bloc-fila sobre el qual actuem
<u>Multiplicar un bloc</u>	<ul style="list-style-type: none"> ▪ Indicar el bloc columna o el bloc fila sobre el que actuem ▪ L'escalar a multiplicar
<u>Pivot per diagonals</u>	<ul style="list-style-type: none"> ▪ Si l'operació és per files o per columnes (*) ▪ Coordenades del bloc de la diagonal D_{ijh} ▪ L'escalar pel qual es multiplica ▪ La diagonal D_{ijh} sobre la qual actuem (destí)
<u>Neteja segons direcció</u>	<ul style="list-style-type: none"> ▪ Les coordenades del bloc origen ▪ La direcció desitjada: dreta, esquerra, amunt o avall
<u>Neteja tot</u>	<ul style="list-style-type: none"> ▪ Les coordenades (fila, columna) de les diagonals principals
<u>MM forma</u>	-

(*) En el cas de pivotar per columnes no cal introduir la fila destí i en el cas de pivotar per files no cal introduir la columna destí (estan inhabilitades)

S'obté la taula de la matriu BLD actualitzada amb cada operació. Al full 2 podem consultar

les propietats de la BLD així com les diagonals utilitzades per obtenir la forma reduïda final, tant si han estat seleccionades com si són les de la MM tipus.

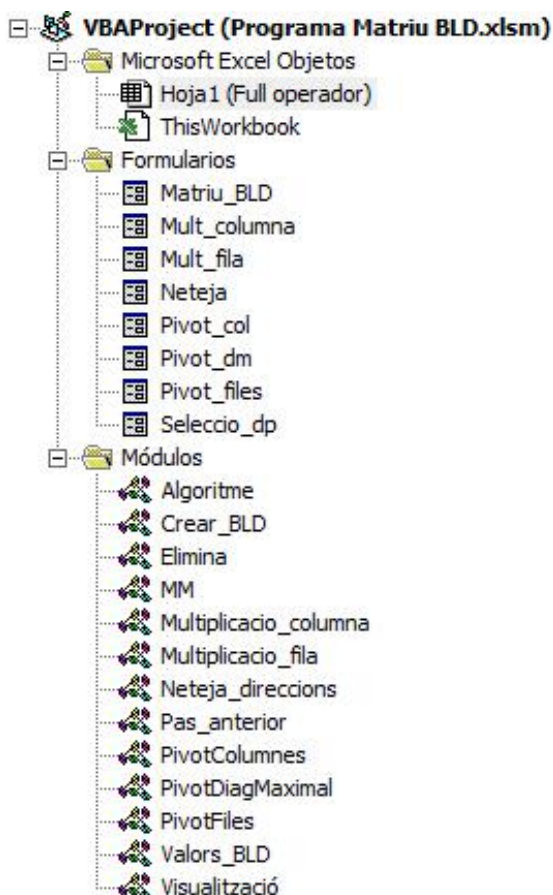
Estructura del programa

El programa està tot emmagatzemat en el document *Programa Matriu BLD*, on hi ha els Objectes, els Formularis i els Mòduls.

Els Objectes és allò que veiem a la pantalla d'execució del programa, és a dir, el *Full operador* amb la Taula d'Instruccions. Aquesta taula té els botons que criden les respectives funcions.

Els Formularis són les diferents interfícies d'entrada de dades, que són cridats per les funcions dels mòduls en executar-se. Més endavant veurem com funcionen.

I finalment els Mòduls són el conjunt de funcions que realitzen les diferents operacions. Vegem-los.



Imatge 2 Estructura del programa

Mòduls

Crear_BLD

El mòdul *Crear_BLD* és el primer en executar-se, crea la base sobre la qual els altres mòduls poden treballar. Conté quatre funcions, la primera funció, *agregar()*, prepara l'espai de treball per poder operar, primerament situa la finestra del *Full operador* per a poder veure-la mentre treballem, depenent del tipus d'Excel. Seguidament crea la nova finestra *Matriu_BLD* i la situa al costat. Prepara l'espai de treball deixant-lo en blanc, escrivint BLD a la casella "A1", assignant amplada i alçada de les caselles i creant i preparant les tres altres fulls del llibre. Finalment crida el Formulari *Matriu_BLD* per a introduir les dades.



La funció *fer_fila*(p, j, x), a partir d'una p , una coordenada d'inici j i el número de fila x , selecciona el conjunt p de caselles lliures, les uneix en una sola i hi escriu el subíndex i la dimensió. D'igual manera la funció *fer_columna*(q, i, x), a partir d'una q , una coordenada d'inici i i el número de columna x , selecciona el conjunt q de caselles lliures, les uneix en una sola i hi escriu el subíndex i la dimensió.

I la quarta funció és *marges_bloc*(i, j, p, q) que, essent:

- i = bloc columna (coordenada horitzontal)
- j = bloc fila (coordenada vertical)
- p = valor p de la fila (alçada)
- q = valor q de la columna (amplada)

Crea els marges d'un bloc de la matriu BLD. Selecciona el bloc des de la casella d'inici de coordenades i i j fins a la de valor $i + q$ i $j + p$, i pinta els marges externs en línia prima i els interns en discontinua.

Valors_BLD

Conté la funció *introduir*() que introdueix els valors de les diagonals de cada bloc. Si el llibre *Matriu_BLD* no està obert dóna error. A partir de les coordenades d'inici (1,1) de la fila 1 i la columna 1 recorre, per cada fila, cada una de les columnes, definint així el recorregut per cada bloc. A cada bloc recorre, per cada subfila, cada una de les subcolumnes, on, si la casella correspon a una diagonal li aplica la fórmula de cridar el valor de la casella de la seva diagonal que es troba a la última subfila del bloc. Si la casella no correspon a cap possible diagonal escriu 0. Les caselles de l'última fila que formen part de les diagonals les pinta d'un altre color (el color pot variar segons la versió d'Excel). Protegeix totes les caselles menys les pintades de color, per a evitar errors. Finalment llença un missatge conforme ha acabat d'executar, informant de la introducció dels valors a les caselles marcades, només cal clicar *D'acord*.

Visualització

Conté les funcions *amagar*() i *descobrir*(). La primera selecciona les files dels blocs excepte la última de cada un i les amaga. La segona selecciona tot el full i el mostra tot, desbloquejant així les files que estiguin amagades. Per poder operar, les dues funcions primer desprotegeixen el full i quan acaben el torna a protegir, però si el llibre *Matriu_BLD* no està obert donen error.

Multiplicacio_columna

Conté la funció *crida_mc*(), que, si està obert el llibre *Matriu_BLD*, crida el formulari *Mult_columna* per introduir els valors, sinó dóna error. I conté també la funció

mult_columnnes(col, esc) que a partir d'una determinada columna la multiplica per l'escalar introduït. Per cada una de les últimes subfiles recorre les caselles que formen part de les diagonals del bloc i multiplica el valor que tenen per l'escalar. Al modificar la última fila les caselles de les diagonals del bloc canvien de valor automàticament. Finalment anota l'operació realitzada al tercer full.

Multiplicacio_fila

Conté la funció *crida_fc()*, que si està obert el llibre *Matriu_BLD*, crida el formulari *Mult_fila* per introduir els valors, sinó dóna error. Conté també la funció *mult_files(fila, esc)* que a partir d'una determinada fila la multiplica per l'escalar introduït. Per cada una de les columnnes recorre les caselles de la última subfila que formen part de les diagonals del bloc i multiplica el valor que tenen per l'escalar. Al modificar la última fila les caselles de les diagonals del bloc canvien de valor automàticament. Finalment anota l'operació realitzada al tercer full.

PivotColumnnes

Per poder introduir els valors, la funció *crida_pc()*, si està obert el llibre *Matriu_BLD*, obre el formulari *Pivot_col*, sinó dóna error. Amb els valors introduïts la funció *Pivot_columnnes(col, esc, dest, numcol)* suma els valors de la columna *col* multiplicats per l'escalar *esc* a la columna *dest*. Per cada fila de la columna destí recorre els valors de la última subfila i aplica l'operació:

$$\text{valor inicial columna dest} + \text{escalar} * \text{valor columna col} = \text{nou valor columna dest}$$

Prenent les *numcol* de més a la dreta del bloc columna *col* i aplicant-se a les *numcol* de més a l'esquerra del bloc columna destí. Finalment anota l'operació realitzada al tercer full.

PivotFiles

La funció *crida_pf()* obre el formulari corresponent per a introduir valors si està obert el llibre *Matriu_BLD*, sinó dóna error. Per crear les files utilitza la funció *Pivotar_files(fila, esc, dest, numfil)* que, essent tots els valors dins del rang permisible, recorre, per cada columna, la última filera de caselles del bloc fila destí. Com l'operació ha de prendre els valors de les files superiors de la fila d'origen (*dest*) i algunes tenen 0s que no es troben a la última subfila, és més fàcil prendre els valors de la primera columna a partir de la coordenada corresponent segons la profunditat desitjada. És a dir que per cada bloc columna recorre de 1 fins al valor mínim *q* o *numfil*, agafant els valors de la primera columna, els multiplica per l'escalar *esc* i els suma als primers valors de la última fila del bloc fila *dest*, ja que opera sobre les files inferiors del bloc destí. Finalment anota l'operació realitzada al tercer full.



PivotDiagMaximal

Conté les funcions *crida_dm()* que si està obert el llibre *Matriu_BLD* crida el formulari corresponent, sinó dóna error, i la funció *Pivotar_dm(col fila, esc, col_f, fila_f, d)* que fa l'operació de pivotar a partir d'una diagonal principal sobre una altra del mateix bloc fila o del mateix bloc columna. Segons si s'ha introduït el valor *col_f* o el valor *fila_f* pivota per files o per columnes, però l'estructura en els dos casos és la mateixa. En el cas del pivot per columnes, si s'ha introduït el valor *d* (opcional) operarà prenent aquest valor com a *numcol*, sinó prendrà la màxima dimensió que pugui prendre. Si la diagonal té menor o igual dimensió que el bloc destí, emmagatzema els valors de la última fila, de dreta a esquerra, en una col·lecció, en cas contrari llença un error de dimensió. Aleshores pivota tot el bloc essent *numcol* la dimensió de la diagonal, és a dir el mínim entre la *p* i la *q* del bloc origen o el valor *d* introduït. Així la diagonal es torna zeros i ara pot pivotar successivament per les diagonals menors. Per fer-ho primer guarda el valor *origen* de la diagonal a partir de la qual operem que s'aplicarà en totes les operacions que fem ja agafarem sempre el triangle de més a la dreta. Per tornar als valors que teníem inicialment per cada transformació successiva de les diagonals necessitem trobar l'escalar adequat, aquests valors ara són diferents perquè hem pivotat per columnes tot el bloc, per això els hem guardat anteriorment, apliquem la fórmula:

$$\text{valor originàri desitjat} = \text{valor actual} + \text{escalar} * \text{valor diagonal maximal}$$

Per trobar l'escalar però, aïllant-lo de la fórmula anterior, si el valor de la diagonal és 0 no podem operar. Entenem que si és 0 aleshores el valor desitjat ja és el valor actual així que no executa res. En el cas, doncs, que aquest sigui diferent de 0, troba l'escalar adequat i crida la funció *Pivotar_columnes(col, esc, col_f, numcol - x)*, on ara *numcol* és una unitat menor, i per cada operació successiva *x* aplica el pivot per columnes a una dimensió menor, des de $x = 0$ fins a $x = \text{numcol} - 1$. Només opera si la diagonal és de dimensió major que 1, ja que sinó no hi ha triangle inferior que arreglar. Per cada una de les transformacions realitzades queden anotades les operacions al tercer full ja que crida altres funcions per realitzar els canvis. Anàlogament pel cas de pivot per files.

Neteja_direccions

Conté les funcions *crida_neteja()*, *dreta(col, fila)*, *esquerra(col, fila)*, *amunt(col, fila)* i *avall(col, fila)*. La primera crida el formulari per escollir l'opció i indicar la fila i columna d'origen si està obert el llibre *Matriu_BLD*, sinó dóna error. Les altres quatre apliquen la funció *Pivotar_columnes(col, esc, dest, numcol)* o *Pivotar_files(fila, esc, dest, numfil)* successivament per cada bloc al que poden aplicar-la, en la direcció indicada. A més les funcions *dreta(col, fila)* i *esquerra(col, fila)* netegen primer el propi bloc i després tots els que poden. Per exemple la funció *dreta(col, fila)*, donat un bloc origen de columna *col* i fila *fila* de dimensions *p* i *q* localitza la diagonal maximal de dimensió *numcol* i en guarda el valor *origen*.

Per cada columna a la dreta de l'origen troba l'escalar adequat i aplica la funció de pivot per columnes. En el primer cas és la pròpia columna, amb la qual opera de $a = 1$ fins a $a = \text{numcol} - 1$ aplicant el tram de diagonal principal adequat en cada cas, $\text{numcol} - a$. Un cop operada la primera columna opera sobre la resta a la seva dreta, fent nul·les totes les diagonals que no ho siguin ja, agafant el tram de diagonal que necessita segons la dimensió en cada cas.

Algoritme

Conté les funcions *crida_sdp()* i *miniversal()*. La primera, si el llibre *Martiu_BLD* està obert, pinta al segon full *Selecció de diagonals principals* i *Fila, Columna*, per poder emmagatzemar els valors que s'introduiran. Per introduir els valors de les diagonals principals crida el formulari *Seleccio_dp*.

La funció *miniversal()* ordena en memòria interna els valors de les files, permetent així poder accedir fàcilment a la de màxim o mínim valor o seguir l'ordre adequat encara que no s'hagin introduït en ordre. Per ordenar-les en l'ordre desitjat fa servir el mètode bombolla [6], que donat un vector $a_1, a_2, a_3, \dots, a_n$:

- Compara a_1 amb a_2 i els intercanvia si $a_1 > a_2$.
- Segueix fins haver comparat tots els valors a_{n-1} amb a_n .
- Repeteix el procés $n - 1$ vegades

Per tal de no afectar els valors de les diagonals en les transformacions següents guarda el valor d'aquestes per poder després restaurar-les amb els seus valors originals.

Per cada diagonal seleccionada recorre:

- De la més avall a la més amunt i aplica *amunt(col, fila)*
- De la més amunt a la més avall, aplica *dreta(col, fila)* i *esquerra(col, fila)*
- De la més avall a la més amunt i aplica *avall(col, fila)*

Finalment restaura els valors de les diagonals amb els valors guardats multiplicant per files.

MM

Conté la funció *tipus()* i la funció *forma()*. La primera selecciona les diagonals per poder obtenir la MM forma, la segona simplement crida l'algoritme de *miniversal()*.

La funció *tipus()*, si està obert el llibre *Matriu_BLD* escriu al full 2 *Diagonals MM Tipus, Fila i*



Columna. Per cada columna recorre cada fila de la més avall a la més amunt i s'atura en aquella que té $p \geq q$ i si no està ja emmagatzemada, l'escriu al full 2. A continuació porta la diagonal principal seleccionada a 1 si aquesta té un valor diferent i aplica *amunt*(col, fila) i *dreta*(col, fila) per netejar.

Pas anterior

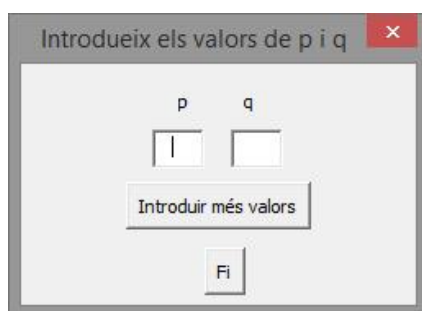
A cada funció que executa operacions es guarden els valors de les últimes subfiles de cada fila i la posició de la última operació en el full 3. Així doncs el botó *Pas anterior* crida la funció *enrere*() que adjudica alter cop els valors guardats i esborra les files d'operacions des de la indicada fins al final. Els valors es guarden en un quart full que queda amagat en tot moment.

Elimina

Activa el llibre "Matriu_BLD" per si no està actiu i el tanca.

Formularis

Matriu_BLD



Imatge 3 Formulari entrades p i q

El botó *Introduir més valors* correspon a la funció *introduir_p_Click()*, que adjudica a p el valor introduït i si s'ha introduït una q l'adjudica també. En cas que el valor de p sigui menor al valor de q introduït llença un missatge d'error i permet corregir el valor. Si és correcte i p diferent de 0 afegeix els valors de p a la llista del full 2. Si q és diferent de 0 l'afegeix també a la segona columna del full 2. Seguidament buida les dues caselles i situa el cursor a la casella de les p altre vegada per seguir introduint més valors.

El botó *Fi* correspon a la funció *Finalitzar_Click()* que per cada valor de les p introduïdes crida la funció *fer_fila*($p, j, fila$), situant j a la coordenada de la última fila no ocupada del full 1. Per cada valor de les q introduïdes crida *fer_columna*($q, i, columna$) situant la coordenada i a la última columna no ocupada. Un cop fetes les files i columnes pinta els marges, per cada fila recorre les diferents columnes i crida la funció dels marges.

Mult_columna

Imatge 4 Formulari entrada dades Mut_columna

El botó *Operar* correspon a la funció *Operar_Click()* que guarda el valor introduït a la casella de *Bloc columna* a la variable *col*. En el cas que no s'introdueixi un valor de l'escalar s'entén que aquest és 1, si el valor introduït és 0 dóna error i en cas contrari el guarda a la variable *esc* per cridar la funció *mult_columnes(col, esc)*. La variable *esc* està declarada com a *Variant* sempre, permetent així l'ús de decimals. En cas que s'introdueixi un decimal fent ús d'un punt com a separador el canvia per una coma per entendre'l correctament.

Mult_fila

Imatge 5 Formulari entrada dades Mult_fila

El botó *Operar* correspon a la funció *Operar_Click()* que guarda el valor introduït a la casella de *Bloc fila* a la variable *fila*. En el cas que no s'introdueixi un valor de l'escalar s'entén que aquest és 1, si el valor introduït és 0 dóna error i en cas contrari el guarda a la variable *esc* per cridar la funció *mult_files(fila, esc)*. La variable *esc* està declarada com a *Variant* sempre, permetent així l'ús de decimals. En cas que s'introdueixi un decimal fent ús d'un punt com a separador el canvia per una coma per entendre'l correctament.

Pivot_col

Imatge 6 Formulari entrada dades Pivot_columnes

El botó *Operar* correspon a la funció *Operar_Click()* que assigna els valors introduïts a les variables corresponents *col*, *numcol*, *esc* i *dest*. En el cas que no s'introdueixi un valor de l'escalar s'entén que aquest és 1, si el valor introduït és 0 dóna error i en cas contrari el guarda a la variable *esc*. En cas que s'introdueixi un decimal fent ús d'un punt com a separador el canvia per una coma per entendre'l correctament. Finalment crida la funció *Pivotar_columnes(col, esc, dest, numcol)*.



Pivot_files

Imatge 7 Formulari entrada dades
Pivot_files

El botó *Operar* correspon a la funció *Operar_Click()* que assigna els valors introduïts a les variables corresponents *fila*, *numfil*, *esc* i *dest*. En el cas que no s'introdueixi un valor de l'escalar s'entén que aquest és 1, si el valor introduït és 0 dóna error i en cas contrari el guarda a la variable *esc*. En cas que s'introdueixi un decimal fent ús d'un punt com a separador el canvia per una coma per entendre'l correctament. Finalment crida la funció *Pivotar_files(files, esc, dest, numfil)*.

Pivot_dm

Imatge 8 Formulari entrada dades
Pivot_dm

Els botons d'opció corresponen a les funcions *OpcioColumnes_Click()* i *OpcioFiles_Click()*, que habiliten la casella de *Bloc columna* de la diagonal sobre la que actuem i inhabiliten la del *Blof fila* si es selecciona l'opció de pivotar per columnes. En cas de seleccionar l'opció de pivotar per files fa el revés. El botó *Operar* correspon a la funció *Operar_Click()* que assigna els valors introduïts a les variables *col*, *fila*, *esc*, *col_f* i *fila_f* si aquests estan dins els rangs permesos. Si no s'introdueix cap escalar li assigna el valor 1, si s'assigna un 0 dóna Error i en cas contrari el guarda a la variable *esc*. En cas que s'introdueixi

un decimal fent ús d'un punt com a separador el canvia per una coma per entendre'l correctament. En el cas que no hi hagi la variable *fila_f* o la variable *col_f*, al estar inhabilitades, les guarda com a 0. Finalment crida la funció *Pivotar_dm(col, fila, esc, col_f, fila_f)* amb les variables assignades.

Neteja

Imatge 9 Formulari entrada dades Neteja

Per cada una de les opcions a escollir crida la corresponent de les funcions *OpcioAvall_Click()*, *OpcioAmunt_Click()*, *OpcioDreta_Click()* i *OpcioEsquerra_Click()* que desmarquen les altres opcions, de manera que només pot estar activa una d'elles. El botó *Neteja* correspon a la funció *Operar_Click()*, que, si no s'ha seleccionat cap de les opcions, dóna error.

Després assigna les variables *col* i *fila* als valors introduïts de la diagonal origen. I si són majors a les dimensions de la BLD dóna Error. Si l'opció escollida és a la dreta però la columna *col* no té més columnes a la dreta dóna error, en cas contrari crida la funció *dreta(col, fila)*. Si l'opció escollida és cap a l'esquerra però la columna *col* no té columnes a l'esquerra també dóna error. D'igual manera succeeix amb les opcions amunt o avall.

Seleccio_dp

Imatge 10 Formulari entrada dades Seleccio_dp

La funció *Introduir_Click()* mira si la fila o la columna que s'introdueix ja ha estat introduïda, en cas contrari l'escriu al full 2 on correspon, en cas que ja s'hagi introduït una fila, una columna o un bloc llença l'error corresponent perquè l'usuari conegui la causa. Si les coordenades indicades estan fora el rang de la matriu BLD també dóna error. A més, mentre s'introdueixen els valors es veuen els que ja es van introduint.

En acabar buida les caselles fila i columna i situa el cursor a la casella de les files per seguir introduint. La funció *Fi_Click()* crida la funció *miniversal()*.



9. Resultats

Per la validació del programa, per la definició dels errors i per l'obtenció dels resultats, s'han anat realitzant un seguit d'exemples, des de matrius de dimensions (1,1) fins a matrius de (15,13), però és vàlid per tota dimensió.

Cada mòdul del programa ha estat validat en diversos tests bàsics, variant les entrades i veient com es desenvolupa el programa a cada pas. El desenvolupador de l'Excel permet veure com evoluciona el programa a cada línia de codi, amb la qual cosa és més fàcil trobar l'error.

Per veure com es comporta el programa en cada cas s'ha treballat amb matrius de dimensions diverses, com:

$$p = (5, 6, 4), q = (4, 6, 3)$$

i

$$p = (4, 4, 4, 6, 3), q = (4, 3, 4, 5, 2)$$

Comprovant així que el programa funciona per configuracions on la p és major en files inferiors i ídem per les columnes, i veient diferents configuracions dels blocs.

Per comprovar l'algoritme s'ha partit d'una matriu BLD amb les dimensions de l'exemple 1.2.9 de la tesi de Marta Peña [1]:

$$p = (6, 4, 3, 2), q = (4, 2, 1)$$

De la qual es coneix la deformació miniversal que s'ha d'obtenir. Amb aquest exemple doncs, vegem els resultats amb diferents valors de les diagonals:

Exemple 9.1 Partint de la matriu

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	4	0	0	0	0	0	0
	3	4	0	0	0	0	0
	2	3	4	0	2	0	0
	1	2	3	4	1	2	3
$p_2 = 4$	1	0	0	0	0	0	0
	2	1	0	0	0	0	0
	3	2	1	0	1	0	0
	4	3	2	1	2	1	1
$p_3 = 3$	3	0	0	0	0	0	0
	2	3	0	0	2	0	0
	1	2	3	0	1	2	3
$p_4 = 2$	1	0	0	0	2	0	0
	2	1	0	0	3	2	1

Imatge 11 Matriu BLD inicial exemple 9.1

Seleccionant les diagonals principals:

Bloc-fila	Bloc-columna
1	1
2	3
3	2

Vegem com evoluciona la matriu en cada un dels passos per l'obtenció de la deformació miniversal. Operem primer per files, pivotant amb cada una de les diagonals principals seleccionades de la de més baix al més amunt (Imatge 12). A continuació, seleccionant les diagonals en l'ordre invers, pivotarem per columnes a dreta i esquerra (Imatge 13).

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	4	0	0	0	0	0	0
	0	4	0	0	0	0	0
	0	0	4	0	0	0	0
	0	0	0	4	0	0	0
$p_2 = 4$	1	0	0	0	0	0	0
	0,5	1	0	0	0	0	0
	-0,3	0,5	1	0	0	0	0
	2	-0,3	0,5	1	0	0	-0,5
$p_3 = 3$	3	0	0	0	0	0	0
	2	3	0	0	2	0	0
	1	2	3	0	1	2	3
$p_4 = 2$	1	0	0	0	2	0	0
	2	1	0	0	3	2	1

Imatge 12 Matriu BLD exemple 9.1, en el primer pas de l'algoritme

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	4	0	0	0	0	0	0
	0	4	0	0	0	0	0
	0	0	4	0	0	0	0
	0	0	0	4	0	0	0
$p_2 = 4$	1	0	0	0	0	0	0
	0,5	1	0	0	0	0	0
	-0,3	0,5	1	0	0	0	0
	0	-0,3	0,5	1	0	0	-0,5
$p_3 = 3$	3	0	0	0	0	0	0
	0	3	0	0	2	0	0
	0	0	3	0	0	2	0
$p_4 = 2$	-1	0	0	0	2	0	0
	-9	-1	0	0	2	2	-2

Imatge 13 Matriu BLD exemple 9.1, en el segon pas de l'algoritme

Altres cops pivotant per files, amb les diagonals principals seleccionades de la de més baix al més amunt (Imatge 14) i finalment, per tornar als valors de les diagonals principals originals multipliquem la fila 2 per -2 i obtenim la deformació miniversal (Imatge 15).

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	4	0	0	0	0	0	0
	0	4	0	0	0	0	0
	0	0	4	0	0	0	0
	0	0	0	4	0	0	0
$p_2 = 4$	1	0	0	0	0	0	0
	0,5	1	0	0	0	0	0
	0	0,5	1	0	0	0	0
	0	0	0,5	1	0	0	-0,5
$p_3 = 3$	3	0	0	0	0	0	0
	0	3	0	0	2	0	0
	0	0	3	0	0	2	0
$p_4 = 2$	-4	0	0	0	2	0	0
	-9	-4	0	0	0	2	-2

Imatge 14 Matriu BLD exemple 9.1, en el tercer pas de l'algoritme

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	4	0	0	0	0	0	0
	0	4	0	0	0	0	0
	0	0	4	0	0	0	0
	0	0	0	4	0	0	0
$p_2 = 4$	-2	0	0	0	0	0	0
	-1	-2	0	0	0	0	0
	0	-1	-2	0	0	0	0
	0	0	-1	-2	0	0	1
$p_3 = 3$	3	0	0	0	0	0	0
	0	3	0	0	2	0	0
	0	0	3	0	0	2	0
$p_4 = 2$	-4	0	0	0	2	0	0
	-9	-4	0	0	0	2	-2

Imatge 15 Matriu BLD exemple 9.1, deformació miniversal

Partint de la mateixa matriu BLD original obtenim la MM tipus i la MM forma (Imatge 16).

Diagonals MM Tipus	
Fila	Columna
2	1
4	2
3	3

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
$p_2 = 4$	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
$p_3 = 3$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	1
$p_4 = 2$	0	0	0	0	1	0	0
	0	0	0	0	0	1	0

Imatge 16 Matriu BLD exemple 9.1, MM forma

Exemple 9.2. Partint d'una matriu de les mateixes dimensions que en l'exemple 9.1, però amb valors de les diagonals diferents (Imatge 17) i a partir de la mateixa selecció de diagonals principals obtenim la deformació miniversal (Imatge 18).

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	1	0	0	0	0	0	0
	0,3	1	0	0	0	0	0
	0,2	0,3	1	0	0,2	0	0
	0,1	0,2	0,3	1	0,1	0,2	0,3
$p_2 = 4$	0,1	0	0	0	0	0	0
	0,2	0,1	0	0	0	0	0
	0,3	0,2	0,1	0	0,1	0	0
	0,4	0,3	0,2	0,1	0,2	0,1	1
$p_3 = 3$	0,3	0	0	0	0	0	0
	0,2	0,3	0	0	1	0	0
	0,1	0,2	0,3	0	0,1	1	0,2
$p_4 = 2$	0,1	0	0	0	0,2	0	0
	0,2	0,1	0	0	0,3	0,2	0,1

Imatge 17 Matriu BLD inicial exemple 9.2

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
$p_2 = 4$	0,1	0	0	0	0	0	0
	0,15	0,1	0	0	0	0	0
	0	0,15	0,1	0	0	0	0
	0	0	0,15	0,1	0	0	1
$p_3 = 3$	0,3	0	0	0	0	0	0
	0	0,3	0	0	1	0	0
	0	0	0,3	0	0	1	0
$p_4 = 2$	-0	0	0	0	0,2	0	0
	0,11	-0	0	0	0	0,2	0,06

Imatge 18 Matriu BLD exemple 9.2, deformació miniversal

I obtenim la MM tipus i la MM forma següents:

Diagonals MM Tipus	
Fila	Columna
2	1
4	2
3	3

BLD	$q_1 = 4$				$q_2 = 2$		$q_3 = 1$
$p_1 = 6$	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
$p_2 = 4$	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
$p_3 = 3$	0	0	0	0	0	0	0
	0	0	0	0	0	0	1
	0	0	0	0	1	0	0
$p_4 = 2$	0	0	0	0	0	1	0
	0	0	0	0	0	0	0

Imatge 19 Matriu BLD exemple 9.2, MM forma

Conclusions

En la primera part del projecte s'han estudiat els subespais invariants i les necessitats de programació. S'ha analitzat quines eren les possibles solucions i finalment es va optar per fer ús del llenguatge Visual Basic en el programador de macros de l'Excel.

L'objectiu era obtenir una eina programada amb un conjunt d'algorismes i una interfície senzilla amb l'objectiu de poder operar amb les matrius BLD i obtenir-ne la deformació miniversal i la seva forma reduïda maximal marcada.

El resultat final de la programació són més de 2.000 línies de codi en Visual Basic d'Excel que, organitzades per mòduls i formularis, permeten anar executant les diferents possibilitats fins arribar a les formes reduïdes maximal marcades.

El programa permet minimitzar l'esforç en la simplificació de les matrius BLD, ja que altrament s'haurien de fer a mà, reduint el temps d'obtenció de les formes reduïdes maximal marcades i operant sense marge d'error humà. A més a més permet comprovar càlculs fets a mà d'una manera fàcil, eficient i segura.

En el desenvolupament del programa s'han pogut veure implementades les operacions de pivot per files i pivot per columnes, amb les restriccions pertinents. A partir d'aquestes transformacions s'ha pogut veure com esdevé el pivot per diagonals, tant per files com per columnes, permetent la neteja successiva.

L'eina obtinguda compleix doncs amb els objectius establerts, per matrius BLD de qualsevol dimensió.

S'ha pogut observar que l'algoritme per l'obtenció de la deformació miniversal podia fer-se amb el pivot per diagonals o pivot per files i columnes, obtenint els mateixos resultats en els dos casos, escollint doncs la metodologia que s'executa més ràpid, amb el pivot per files i columnes ja que conté menys operacions.

En els resultats s'ha estudiat a partir de l'exemple 1.2.9 de [1] de la qual s'ha obtingut la deformació miniversal correcta. Finalment s'ha pogut veure que s'obté la mateixa forma reduïda maximal marcada per matrius BLD de mateixes dimensions però amb valors diferents, confirmant així la teoria.

El projecte permet avançar en l'estudi de la classificació dels subespais invariants, oferint una eina ràpida i senzilla que demostra la teoria. Deixa oberts també altres possibles estudis i millores del programa, com:

- L'obtenció de les matrius P i Q per cada transformació

- Reduir les diagonals maximals a 1 amb triangle inferior nul abans de pivotar
- Seguir simplificant la forma reduïda trobada amb pivots addicionals
- Obtenir l'arbre de les pertorbacions possibles
- Detecció d'errors del programa, com en usos d'Excel de diferents versions

Agraïments

Voldria fer agraïment a totes les persones que m'han acompanyat en la realització d'aquest projecte.

Agraeixo al meu tutor, Josep Ferrer, per acceptar ser el meu tutor, proposar-me el treball, per la gran ajuda i la paciència de treballar a distància.

A la Marta Peña, per l'ajuda i la paciència durant l'elaboració del projecte, i per demostrar que l'enginyeria i l'àlgebra lineal s'uneixen.

Al Departament de Matemàtica Aplicada I per permetrem la possibilitat de formar part del projecte i facilitar-me tota la informació.

A les persones que m'han acompanyat durant el projecte de cooperació internacional que m'han permès tenir una dedicació parcial per poder dedicar-me als dos projectes alhora.

Als meus amics i amigues per la comprensió i interès pel projecte en tot el que han pogut.

A la meva família per confiar en mi i mostrar-me el seu orgull davant la meva tasca.

Bibliografia

En el conjunt de la memòria es fa referència als diversos treballs publicats pel Departament de Matemàtica Aplicada I de la UPC, els quals es citen a continuació, i s'ha fet ús de diverses fonts d'internet per l'aprenentatge del llenguatge de programació Visual Basic.

Referències bibliogràfiques

- [1] M. Peña Carrera / *Deformacions d'operadors i subespais associats a sistemes lineals multivariats* / Tesi Doctoral, Universitat Politècnica de Catalunya, 2005.
- [2] A. Compta / *Contribució a l'estudi geomètric de subespais invariants respecte a transformacions i sistemes lineals* / Tesi Doctoral, Universitat Politècnica de Catalunya, 2001
- [3] A. Compta; J. Ferrer; M. Peña / *Dimension of the Orbit of Marked Subspaces* / Linear Algebra Appl., 2004.
- [4] J. Ferrer; F. Puerta / *Versal Deformations of Invariants Subspaces* / Linear Algebra Appl., 2001.
- [5] A. Compta; J. Ferrer; F. Puerta / *Miniversal Deformations of Marked Matrices* / Linear Algebra Appl., 2003
- [6] [<http://www.estructuradedatos.galeon.com/burbujatext.htm>, 13 de març de 2016]: Teoria del mètode de la bombolla d'ordenament de dades.

Bibliografia complementària

- [7] Excel Total [<https://exceltotal.com/>]: Consells, guies i tutorials sobre l'ús avançat d'Excel.
- [8] Excel Avanzado [<http://www.excel-avanzado.com/>]: Conjunt de casos i tutorials sobre els macros d'Excel.
- [9] Catàleg de referència i API de Microsoft [<https://msdn.microsoft.com/library>]: Explicacions, exemples i recursos de l'entorn de desenvolupament Visual Studio com el Visual Basic.



- [10]** Excel Easy [<http://www.excel-easy.com/vba.html>]: Tutorials i exemples sobre l'ús de l'Excel
- [11]** Ajuda Microsoft [<https://support.microsoft.com>]: Plataforma d'ajuda de Microsoft, manual